IN THE CLAIMS:

Please amend the claims as follows:

1.   (Currently Amended)   A method for executing location independent procedure calls in a network system, comprising:

determining a remote node to execute a function, if the function cannot be run on a local node;

executing a route process ~~function~~, wherein the route process ~~function~~ comprises:

generating a descriptive data structure for parametric function-related data;

generating a pure value buffer <u>derived from the parametric function-related data</u>; and

flattening the <u>parametric function-related</u> data ~~structure~~ and the pure value buffer into a bundle;

transmitting the flattened data ~~structure~~ and descriptive data to the remote node;

executing <u>the</u> function on the remote node; and

transmitting the results of the function to the local node.

2.   (Previously Presented)   The method of claim 1, wherein generating a descriptive data structure comprises generating a DTSTRUCT.

3.   (Currently Amended)   The method of claim 1 ~~2~~, wherein the pure value buffer comprises a data-only <u>buffer</u>.

4.   (Previously Presented)   The method of claim 1, further comprising:

determining if the bundle is cacheable;

determining if the bundle is available in cache memory if it is determined to be cacheable; and

276734_1

retrieving a cached reply from the cache memory if the bundle is determined to be cacheable and available in cache memory.

5.   (Original) The method of claim 1, wherein determining a remote node to execute a function further comprises reading a parameter associated with the function, wherein the parameter associated with the function indicates where the function may be executed.

6.   (Previously Presented)   The method of claim 1, wherein flattening comprises assembling each variable argument indicated in the route function into a buffer.

7.   (Previously Presented)   The method of claim 1, wherein executing the function on the remote node further comprises:
        receiving the bundle on the remote node;
        unpackaging the bundle on the remote node;
        computing the function on the remote node; and
        packaging a function reply.

8.   (Previously Presented)   The method of claim 7, wherein packaging a function reply further comprises flattening the function reply.

9.   (Previously Presented)   The method of claim 7, wherein unpackaging the bundle further comprises unflattening the bundle.

10.   (Previously Presented)   The method of claim 8, further comprising:
        receiving the transmitted results of the function on the local node;
        determining if the transmitted results are cacheable; and
        storing the transmitted results in a cache memory if the transmitted results are determined to be cacheable.

11.   (Original) The method of claim 1, further comprising:

queuing at least one of pre-flattened commands and flattened commands prior to transmission to a remote node; and

cooperatively executing the queued commands in a single network transaction.

12. (Currently Amended) A method for transparently executing function calls from a local node on a remote node, comprising:

determining a remote node for execution of a function <u>call</u>;

calling a route function configured to generate a flattened pure value buffer containing <u>parametric</u> function-related data;

transmitting the <u>flattened pure value</u> buffer from the local node to the remote node;

executing the function <u>call</u> on the remote node; and

transmitting results of the function <u>call</u> to the local node.

13. (Currently Amended) The method of claim 12, wherein determining a remote node further comprises reading a parameter associated with the function <u>call</u>, wherein the parameter indicates the remote node for execution of the function <u>call</u>.

14. (Currently Amended) The method of claim 12, wherein calling a route function further comprises:

generating a text string, wherein each element of the text string identifies the data type of a portion of the <u>parametric</u> function-related data; and

bundling the <u>parametric</u> function-related data.

15. (Original) The method of claim 14, wherein the text string further comprises a DTSTRUCT string.

16. (Currently Amended) The method of claim 14, wherein bundling further comprises flattening the <u>parametric</u> function-related data.

Page 4

276734_1

PAGE 5/15 * RCVD AT 5/26/2004 6:04:34 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:+7136234846 * DURATION (mm-ss):04-28

17. (Currently Amended)   The method of claim 12, wherein executing the function call on the remote node further comprises:

unbundling the parametric function-related data;

computing the function call; and

bundling reply to the function call.

18. (Original) The method of claim 17, wherein unbundling and bundling further comprise unflattening and flattening, respectively.

19. (Currently Amended) The method of claim 17, further comprising looking up a function pointer that indicates the location of the function call to the remote node.

20. (Currently Amended) The method of claim 12, further comprising determining if the parametric function related data is cacheable and storing the parametric function-related data in cache memory if the parametric function-related data is determined to be cacheable.

21. (Currently Amended)   The method of claim 12, further comprising determining if the results of the function call are cacheable and storing the results of the function call in cache memory if the results of the function call are determined to be cacheable.

22. (Currently Amended)   The method of claim 12, wherein transmitting the results of the function call to the local node further comprises unflattening the results.

23. (Previously Presented)   The method of claim 12, further comprising:
queuing at least one of pre-flattened commands and flattened commands prior to transmission to the remote node; and
cooperatively executing the queued commands in a single network transaction.

Page 5

276734_1

PAGE 6/15 * RCVD AT 5/26/2004 6:04:34 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:+7136234846 * DURATION (mm-ss):04-28

24.   (Currently Amended)   A computer readable medium storing a software program that, when executed by a processor, causes the processor to perform a method comprising:

determining a remote node to execute a function, if the function cannot be run on a local node;

executing a route process ~~function~~ configured to assemble a flattened pure value buffer containing parametric function-related data;

transmitting the flattened pure value buffer to the remote node;

executing the function on the remote node; and

transmitting the results of the function to the local node.

25.   (Currently Amended)   The computer readable medium of claim 24, wherein executing a route process ~~function~~ further comprises:

generating a parameter representative of the parametric function-related data; and

packaging the parametric function-related data and the generated parameter for transmission to the remote node.

26.   (Currently Amended)   The computer readable medium of claim 25, wherein the parameter representative of the parametric function-related data further comprises a text string, wherein each character in the text string corresponds to a particular data type.

27.   (Currently Amended)   The computer readable medium of claim 25, further comprising:

determining if the packaged function-related data is cacheable;

determining if the packaged function-related data is available in cache memory if it is determined to be cacheable; and

retrieving a cached reply from the cache memory of the packaged function-related data is determined to be cacheable and available in cache memory.

28.   (Original) The computer readable medium of claim 24, wherein determining a remote node to execute a function further comprises reading a parameter associated with the function, wherein the parameter associated with the function indicates where the function may be executed.

29.   (Currently Amended) The computer readable medium of claim 25, wherein packaging the function related data and the generated parameter further comprises flattening each variable argument indicated in the route function into a ~~the pure value~~ buffer.

30.   (Currently Amended) The computer readable medium of claim 24, wherein executing the function on the remote node further comprises:
        receiving the <u>parametric</u> function and related data on the remote node;
        unpackaging the <u>parametric</u> function-related data on the remote node;
        computing the function on the remote node; and
        packaging a function reply.

31.   (Original) The computer readable medium of claim 30, wherein packaging a function reply further comprises flattening the reply.

32.   (Currently Amended) The computer readable medium of claim 30, wherein unpackaging the <u>parametric</u> function-related data further comprises unflattening the function related data.

33.   (Original) The computer readable medium of claim 24, further comprising:
        receiving the transmitted results of the function on the local node;
        determining if the transmitted results are cacheable; and
        storing the transmitted results in a cache memory if the transmitted results are determined to be cacheable.

34.   (Original) The computer readable medium of claim 24, further comprising:

Page 7

276734_1

queuing at least one of pre-flattened commands and flattened commands prior to transmission to a remote node; and

cooperatively executing the queued commands in a single network transaction.

35.   (Currently Amended)   A computer readable medium storing a software program that, when executed by a processor, causes the processor to perform a method comprising:

determining a remote node for execution of a function call;

calling a route function;

generating a flattened pure value buffer containing parametric function-related data and a descriptive data structure;

transmitting the buffer from the local node to the remote node;

executing the function call on the remote node; and

transmitting results of the function call to the local node.

36.   (Currently Amended)   The computer readable medium of claim 35, wherein determining a remote node further comprises reading a parameter associated with the function call, wherein the parameter indicates the remote node for execution of the function call.

37.   (Original) The computer readable medium of claim 35, wherein calling a route function further comprises:

generating a text string, wherein each element of the text string identifies the data type of a portion of the function related data; and

bundling the function related data.

38.   (Original) The computer readable medium of claim 37, wherein the text string further comprises a DTSTRUCT string.

39.   (Original) The computer readable medium of claim 37, wherein bundling further comprises flattening the function related data.

Page 8

276734_1

PAGE 9/15 * RCVD AT 5/26/2004 6:04:34 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:+7136234846 * DURATION (mm-ss):04-28

40.   (Currently Amended)   The computer readable medium of claim 35, wherein executing the function call on the remote node further comprises:

unbundling the parametric function-related data;

computing the function call; and

bundling the reply to the function call.

41.   (Original) The computer readable medium of claim 40, wherein unbundling and bundling further comprise unflattening and flattening, respectively.

42.   (Currently Amended)   The computer readable medium of claim 40, further comprising looking up a function pointer that indicates the location of the function call to the remote node.

43.   (Currently Amended)   The computer readable medium of claim 35, further comprising determining if the parametric function-related data is cacheable and storing the parametric function-related data in cache memory if the parametric function-related data is determined to be cacheable.

44.   (Currently Amended)   The computer readable medium of claim 35, further comprising determining if the results of the function call are cacheable and storing the results of the function call in cache memory if is the results of the function call are determined to be cacheable.

45.   (Currently Amended)   The computer readable medium of claim 35, wherein transmitting the results of the function call to the remote node further comprises unflattening the results.

276734_1

46.   (Original) The computer readable medium of claim 35, comprising:

queuing at least one of pre-flattened commands and flattened commands prior to transmission to a remote node; and

cooperatively executing the queued commands in a single network transaction.

276734_1